Ιόνιο Πανεπιστήμιο, Τμήμα Πληροφορικής

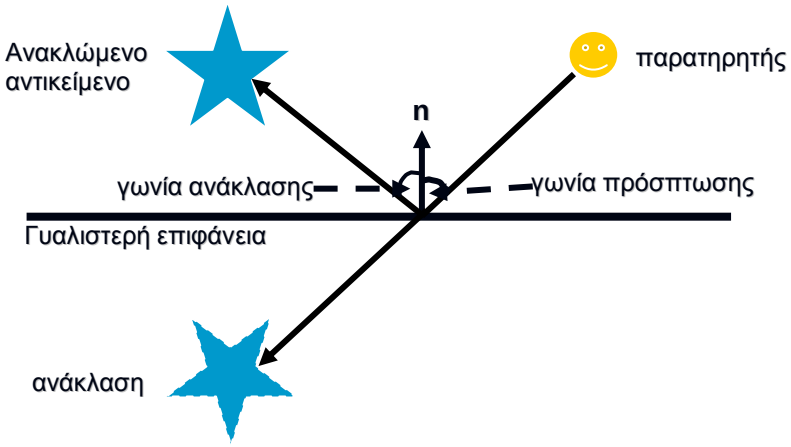
Γραφικά με Υπολογιστές

Εργαστήριο 5 – Σκιές & Ανακλάσεις

Στο εργαστήριο αυτό θα δούμε το πώς εφαρμόζουμε και μπορούμε να δημιουργήσουμε σκιές και ανακλάσεις σε αντικείμενα στην OpenGL.

Ανακλάσεις στην OpenGL

Μια απλή μέθοδος για να δημιουργήσουμε την ανάκλαση ενός αντικειμένου ως προς μια επίπεδη επιφάνεια είναι να μετασχηματίσουμε συμμετρικά το αντικείμενο ως προς την επιφάνεια αυτή και να το απεικονίσουμε.



Αν υποθέσουμε ότι το επίπεδο ανάκλασης περνά από τη αρχή των αξόνων και βρίσκεται πάνω στο επίπεδο X-Z του συστήματος αναφοράς, τότε ο μετασχηματισμός που χρησιμοποιούμε είναι μια απλή κλίμακα $S_{xyz}(1,-1,1)$. Στην OpenGL αυτό γίνεται με την χρήση της εντολής:

```
glScalef(1.0f,-1.0f,1.0f);
```

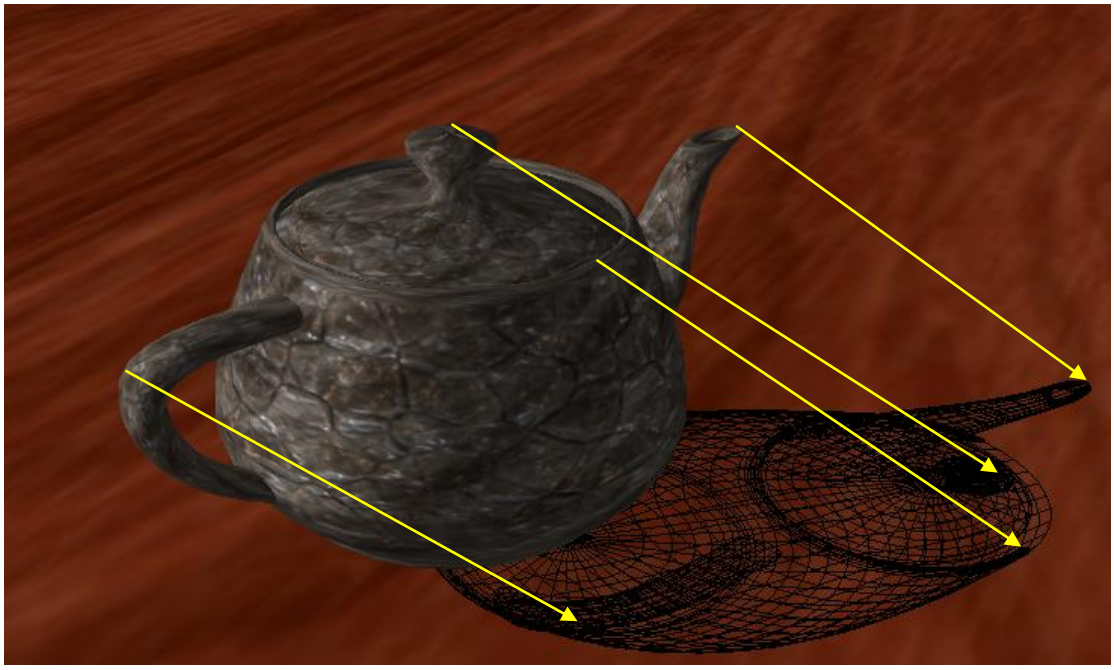
Ο τρόπος με τον οποίο δημιουργούμε την ανάκλαση ενός αντικειμένου είναι ο εξής:

1. Μετασχηματίζουμε το αντικείμενο με την glScalef(1.0f,-1.0f,1.0f) και το απεικονίζουμε. Δημιουργείται έτσι η ανάκλαση.
2. Κάνουμε την επιφάνεια ανάκλασης ημιδιάφανη και την απεικονίζουμε(πάτωμα, καθρέπτης κλπ)
3. Απεικονίζουμε το αντικείμενο στην κανονική του θέση

Ο λόγος που κάνουμε την επιφάνεια ανάκλασης ημιδιάφανη είναι γιατί σε διαφορετική περίπτωση δεν θα φαινόταν η ανάκλαση του αντικειμένου.

Σκιές με προβολή

Η σκιά με προβολή δημιουργείται προβάλλοντας την γεωμετρία ενός αντικειμένου (δηλαδή κάθε κορυφή) στο επίπεδο με τον μετασχηματισμό προβολής από τον χώρο της κάμερα στο χώρο της οθόνης που γνωρίσαμε στο Εργαστήριο 2. Το αντικείμενο της σκιάς έχει όσα πολύγωνα όσα και το αρχικό αντικείμενο.



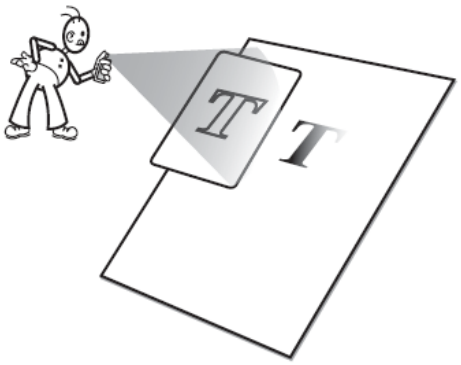
Για να διευκολυνθούμε κατά την κατασκευή αυτού του μετασχηματισμού (projection) θα χρησιμοποιήσουμε μια εντολή που τον υπολογίζει άμεσα δεδομένου την τοποθεσία του φωτός και του επιπέδου που θα «πέσει» η σκιά (η συνάρτηση δεν αποτελεί μέρος της OpenGL ή του GLUT).

```
m3dMakePlanarShadowMatrix(M3DMatrix44f projection, const M3DVector4f
plane, const M3DVector3f vLightPos)
```

Ο stencil buffer

Ο stencil buffer είναι χρήσιμο εργαλείο που μπορούμε να χρησιμοποιήσουμε κατά την κατασκευή ανακλάσεων ή σκιών. Είναι μια περιοχή στην μνήμη με ανάλυση όση και ο buffer στο οποίον απεικονίζουμε την σκηνή μας (συνήθως όση ανάλυση έχει το παράθυρο στο οποίο απεικονίζουμε την σκηνή). Κάθε pixel του stencil buffer μπορεί να κρατά 1 bit (ή και παραπάνω, εξαρτάται από την υλοποίηση του κατασκευαστή του επεξεργαστή γραφικών).

Ο stencil buffer χρησιμοποιείται με την ίδια λογική που τον χρησιμοποιούμε στην πραγματική ζωή: θέτει τα όρια ενός αντικειμένου όταν βάφουμε κάτι με σπρέι.



Με παρόμοιο τρόπο, ανάλογα με την τιμή κάθε pixel του stencil buffer μπορούμε να διαλέξουμε αν θα απεικονίσουμε ένα pixel στην τελική εικόνα ή όχι.

Η χρήση του stencil buffer είναι λίγο περίπλοκη και δεν θα πούμε σε βάθος σε αυτό το εργαστήριο. Για να αρχικοποιήσουμε τον stencil buffer σε κάποια τιμή χρησιμοποιούμε την εντολή

```
glClear(GL_STENCIL_BUFFER_BIT);
```

Αφού προηγουμένως έχουμε θέση την τιμή αρχικοποίησης με την εντολή:

```
glClearStencil(GLint s);
```

Το τεστ που πραγματοποιείται σε κάθε pixel για να αποφασιστεί αν αυτό θα εμφανιστεί στην τελική εικόνα το ορίζουμε με την εντολή:

```
void glStencilFunc(GLenum func, GLint ref, GLuint mask);
```

Το τι θα συμβεί στο stencil buffer ανάλογα με το αποτέλεσμα του τεστ ορίζεται με την εντολή.

```
void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass);
```

Τέλος ο stencil buffer ενεργοποιείται με την χρήση της εντολής.

```
glEnable(GL_STENCIL_TEST);
```

Θα δούμε αργότερα ένα απλό παράδειγμα χρήσης του stencil buffer στην πράξη.

Εφαρμογή σκιών και ανακλάσεων στην OpenGL

Στην σημερινή εργαστηριακή άσκηση θα δημιουργήσουμε την ανάκλαση και την σκιά ενός αντικειμένου στην OpenGL χρησιμοποιώντας τις μεθόδους που αναφέραμε στις άνω παραγράφους

Φορτώστε το project lab5.dev στο περιβάλλον ανάπτυξης εφαρμογών DevC++.

Ο κώδικας της εφαρμογής βρίσκεται στο αρχείο main.cpp. Ο κώδικας περιέχει σχόλια για όλα τα σημεία του προγράμματος που μας ενδιαφέρουν. Κομμάτια του κώδικα και άλλες παραμετροποιήσεις δεν μας αφορούν σε αυτή την άσκηση και μένουν ασχολίαστα.

Η εφαρμογή ακολουθεί το γνωστό σκελετό που χρησιμοποιούμε στις εργαστηριακές ασκήσεις. Οι αλλαγές στην σημερινή άσκηση εστιάζονται στην συνάρτηση αρχικοποίσης init() και στην συνάρτηση που ζωγραφίζει την σκηνή μας renderScene():

init()

Χρησιμοποιούμε 2 βοηθητικές εντολές για να υπολογίσουμε τον μετασχηματισμό προβολής της σκιάς:

```
m3dGetPlaneEquation(pPlane, vPoints[0], vPoints[1], vPoints[2]);
m3dMakePlanarShadowMatrix(mShadowMatrix, pPlane, lightPos);
```

Ο πίνακας 4x4 mShadowMatrix θα κρατά τον μετασχηματισμό ο οποίος δεν αλλάζει καθ'όλη την διάρκεια της εφαρμογής (δηλαδή υποθέτουμε ότι το φως και το επίπεδο προβολή της σκιάς δεν αλλάζουν θέση)

Επίσης στην init() ορίζουμε ότι η αρχική τιμή του stencil buffer θα είναι μηδέν.

```
glClearStencil(0.0f);
```

renderScene()

Στην renderScene() θα ζωγραφίσουμε την σκηνή με την ανάκλαση και την σκιά.
Για να ζωγραφίσουμε την ανάκλαση της τσαγέρας ακολουθούμε τα βήματα που αναφέραμε παραπάνω:
Απεικονίζουμε την σφαίρα αφότου έχουμε «ανακλάσει» τις συντεταγμένες της. Η ανάκλαση θα γίνει στο επίπεδο XZ οπότε πρέπει να βρούμε τον αντίθετο των Y συντεταγμένων και αυτό γίνεται με την συνάρτηση

```
glScalef(1.0f, -1.0f, 1.0f);
```

```
// ΤΜΗΜΑ ΖΩΓΡΑΦΙΖΕΙ ΤΗ ΑΝΑΚΛΑΣΗ ΤΗΣ ΣΦΑΙΡΑΣ *****
glPushMatrix();
      // βρες τον αντίθετο της Y συντεταγμένης
      glScalef(1.0f, -1.0f, 1.0f);
      //περίστρεψε και μετακίνησε την σφαίρα στην επιθυμητή θέση
      glTranslatef(0,ballHeight,0);
      glRotatef(a,0,1,0);
      //ενεργοποίησε την υφή της τσαγέρας
      glBindTexture(GL_TEXTURE_2D, textures[TEXTURE_SPHERE]);
      glutSolidTeapot(4.0f);
glPopMatrix();
```

Στην συνέχεια ζωγραφίζουμε το επίπεδο ορίζοντας ένα ποσοστό 30% διαφάνειας. Ο λόγος που το κάνουμε αυτό είναι διότι μιας και ζωγραφίζουμε το επίπεδο «πάνω» από το αντικείμενο της ανάκλασης θα το κρύβαμε αν το επίπεδο δεν ήταν ημιδιαφανές.

```
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
glColor4f(1.0f, 1.0f, 1.0f,0.7f);
DrawGround();
```

Η glBlendFunc() ορίζει την συνάρτηση που θα χρησιμοποιήσουμε κατά την «μίξη» του χρώματος της ανάκλασης με το χρώμα του επιπέδου. Στην συγκεκριμένη περίπτωση χρησιμοποιούμε την συνάρτηση

$$c_o = a_s c_s + (1-a_s) c_d$$

όπου a_s είναι η τιμή alpha του επιπέδου (που ορίσαμε σε 0.7), c_d το χρώμα της ανάκλασης, c_s το χρώμα του επιπέδου και c_o το αποτέλεσμα της μίξης των 2 αυτών χρωμάτων.

Έπειτα ζωγραφίζουμε την σκιά της τσαγέρας χρησιμοποιώντας την προβολή που υπολογίσαμε στην init()

```
glPushMatrix();
      glColor4f(0.0,0.0,0.0,0.0);
      glMultMatrixf(mShadowMatrix);
      glTranslatef(0,ballHeight,0);
      glRotatef(a,0,1,0);
      glutSolidTeapot(4.0f);
glPopMatrix();
```

Στο τέλος ζωγραφίζουμε και την τσαγέρα

```
glPushMatrix();
      glTranslatef(0,ballHeight,0);
      glRotatef(a,0,1,0);
      glBindTexture(GL_TEXTURE_2D, textures[TEXTURE_SPHERE]);
      glutSolidTeapot(4.0f);
glPopMatrix();
```

Πράγματα να δοκιμάσετε

Α) Στο τμήμα που ζωγραφίζει το επίπεδο (*συνάρτηση renderscene()*) αλλάξτε την τιμή του alpha στην συνάρτηση glColor4f(1.0f, 1.0f, 1.0f,0.7f); Δοκιμάστε μερικές τιμές από 0.0f μέχρι 1.0f.

Τι συμβαίνει στην ανάκλαση και τι στο επίπεδο;

Β) Το τμήμα που ζωγραφίζει την σκιά της τσαγέρας (συνάρτηση renderscene()), δοκιμάστε να ζωγραφίσετε την τσαγέρα με την εντολή glutWireTeapot(4.0f); αντί τις glutSolidTeapot(4.0f);

Τι παρατηρείτε για την σκιά του αντικειμένου;

Γ) Στην συνάρτηση DrawGround() αλλάξτε το μέγεθος του επιπέδου μέσω των μεταβλητών width, length σε 10x10.

Τρέξτε το πρόγραμμα. Τι παρατηρείτε για την σκιά του αντικειμένου;

Μετακινήστε την σκηνή με τα πλήκτρα βελάκια μέχρι να μπορείτε να δείτε κάτω από το επίπεδο. Τι παρατηρείτε για την ανάκλαση του αντικειμένου;

Δ) Ο λόγος που παρατηρούμε τα φαινόμενα του βήματος Γ είναι ότι επιτρέπουμε την σκιά και την ανάκλαση του αντικειμένου να «απεικονιστούν» οπουδήποτε. Αν η κάμερα τύχει να βρίσκεται στην σωστή θέση (πάνω από το επίπεδο) και το επίπεδο είναι αρκετά μεγάλο τότε φαίνονται σωστά, σε διαφορετική περίπτωση το εφε χαλάει.

Από την εμπειρία μας μια ανάκλαση σε μια επιφάνεια φαίνεται μόνο μέσα από την επιφάνεια αυτή. Επίσης μια σκιά που πέφτει σε μια επιφάνεια πέφτει μόνο σ'αυτή (κυρίως, εκτός και αν έχει άλλες επιφάνειες δίπλα της).

Στην περίπτωση μας θα θέλαμε η ανάκλαση να φαίνεται μόνο μέσα από το επίπεδο και η σκιά να πέφτει μόνο πάνω σ'αυτό. Οτιδήποτε πέφτει έξω από το επίπεδο θα

θέλαμε να αποκόπτεται. Ο τρόπος που μπορούμε να το υλοποιήσουμε αυτό είναι μέσω του stencil buffer.

Αρχικά πρέπει να φτιάξουμε το stencil. Δηλαδή να ορίσουμε την περιοχή μέσα στην οποία θα φαίνονται η ανάκλαση και η σκιά και μόνο. Αυτό γίνεται το κομμάτι του κώδικα:

```
glEnable(GL_STENCIL_TEST);
glStencilFunc(GL_ALWAYS, 1, 1);
glStencilOp(GL_KEEP, GL_KEEP, GL_REPLACE);
//ζωγράφισε το επίπεδο στο stencil buffer
DrawGround();
```

Αρχικά ενεργοποιούμε το stencil buffer. Ορίζουμε το τεστ για το stencil buffer ως

```
glStencilFunc(GL_ALWAYS, 1, 1);
```

Η εντολή αυτή λέει: μην αποκόψεις τίποτα από την εικόνα, ζωγράφισε τα πάντα (το τεστ περνά πάντα - `GL_ALWAYS`). Επίσης η εντολή ορίζει ότι σε κάθε pixel που πρόκειται να ζωγραφιστεί στην οθόνη, θέσε την αντίστοιχη τιμή του stencil buffer σε 1.

Ορίζουμε το τι θα συμβαίνει στο stencil buffer κάθε φορά που το τεστ πετυχαίνει με την εντολή `glStencilOp(GL_KEEP, GL_KEEP, GL_REPLACE);` Η τρίτη παράμετρος μας ενδιαφέρει σε αυτή την περίπτωση που ορίζει πως θα μεταβάλλεται η αντίστοιχη τιμή του stencil buffer κάθε φορά που πετυχαίνει το τεστ. Το `GL_REPLACE` ορίζει ότι θα θέσουμε την τιμή του stencil buffer σε 1 (την τιμή 1 την ορίσαμε με την `glStencilFunc() )`.

Στην συνέχεια ζωγραφίζουμε το επίπεδο με την DrawGround(). Αυτό θα έχει σαν αποτέλεσμα ότι για κάθε pixel του επιπέδου στην τελική εικόνα θα υπάρχει μια τιμή 1 στο stencil buffer. Όπου δεν υπάρχει το επίπεδο ο stencil buffer θα έχει τιμή 0.

Με αυτή την πληροφορία (δηλαδή που υπάρχει επίπεδο και που όχι) μπορούμε να ζωγραφίσουμε την ανάκλαση και την σκιά, αρκεί να αλλάξουμε λίγο το τεστ του stencil buffer:

```
glStencilFunc(GL_EQUAL, 1, 1);
glStencilOp(GL_KEEP, GL_KEEP, GL_KEEP);
```

Ορίζουμε ότι θέλουμε να ζωγραφίσουμε την σκιά και την ανάκλαση όπου το stencil buffer έχει τιμή ένα (`GL_EQUAL`) και ότι δεν θέλουμε να μεταβάλλουμε το περιεχόμενο του stencil buffer (`GL_KEEP`).

Δοκιμάστε:

Κατασκευάστε τον stencil buffer πριν ζωγραφίσετε οτιδήποτε στην σκηνή τοποθετώντας το παρακάτω κομμάτι κώδικα

```
glDisable(GL_DEPTH_TEST);
glEnable(GL_STENCIL_TEST);
glStencilFunc(GL_ALWAYS, 1, 1);
glStencilOp(GL_KEEP, GL_KEEP, GL_REPLACE);
// ζωγράφισε το επίπεδο στο stencil buffer
DrawGround();

glStencilFunc(GL_EQUAL, 1, 1);
glStencilOp(GL_KEEP, GL_KEEP, GL_KEEP);
```

στην renderScene() στην θέση

```
/*
      Τοποθετήστε εδώ το κώδικα για τη κατασκευή stencil
*/
```

Τρέξτε το πρόγραμμα. Τι παρατηρείτε τώρα για την σκιά και την ανάκλαση; Τι παρατηρείτε για την τσαγέρα;

Ε) Δεν θέλουμε να χρησιμοποιήσουμε το stencil καθώς ζωγραφίζουμε τη «πραγματική» τσαγιέρα, γιαυτό το απενεργοποιούμε προσθέτοντας την εντολή `glDisable(GL_STENCIL_TEST);` στην αρχή του τμήματος που τοποθετεί, περιστρέφει και ζωγραφίζει την τσαγέρα (την κανονική και όχι την ανάκλαση ή σκιά) στην συνάρτηση renderScene().